View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by ZENODO

$VISA^3$: REFINING THE VOICE INTEGRATION/SEGREGATION ALGORITHM

Dimos Makris
Dept. of Informatics,
Ionian University,
Greece
c12makr@ionio.gr

Ioannis Karydis
Dept. of Informatics,
Ionian University,
Greece
karydis@ionio.gr

Emilios Cambouropoulos
Dept. of Musical Studies,
Aristotle University of
Thessaloniki, Greece
emilios@mus.auth.gr

ABSTRACT

Human music listeners are capable of identifying multiple 'voices' in musical content. This capability of grouping notes of polyphonic musical content into entities is of great importance for numerous processes of the Music Information Research domain, most notably for the better understanding of the underlying musical content's score. Accordingly, we present the $VISA^3$ algorithm, a refinement of the family of VISA algorithms for integration/segregation of voice/streams focusing on musical streams. $VISA^3$ builds upon its previous editions by introduction of new characteristics that adhere to previously unused general perceptual principles, address assignment errors that accumulate affecting the precision and tackle more generic musical content. Moreover, a new small dataset with human-expert ground-truth quantised symbolic data annotation is utilised. Experimental results indicate the significant performance amelioration the proposed algorithm achieves in relation to its predecessors. The increase in precision is evident for both the dataset of the previous editions as well as for a new dataset that includes musical content with characteristics such that of non-parallel motion that are common and have not yet been examined.

1. INTRODUCTION

It is a common understanding of music listeners that musical content can be separated to multiple 'voices'. Nevertheless, it is widely accepted [1–3] that the notion of a 'voice' is far from well-defined as it features in a plethora of alternative meanings, especially when polyphonic and homophonic elements are included.

In most occasions, the term 'voice' refers to a monophonic sequence of successive non-overlapping musical tones, as a single voice is assumed not to contain multi-tone sonorities. In some cases though, provided that 'voice' is examined in the light of auditory streaming, it is possible that the standard meaning is insufficient. In these cases, a single monophonic sequence may be perceived as more than one voices/streams (e.g., pseudopolyphony or implied polyphony) while a sequence containing concurrent notes may be perceived as a single perceptual entity (e.g., homophonic passages) [4].

Musical auditory stream integration/segregation defines how successions of musical events are perceived to be coherent sequences and, at the same time, segregated from other independent musical sequences. A number of general perceptual principles govern the way musical events are grouped together in musical streams [1, 2].

Given the ambiguity of 'voice' segregation definition, the process can be separated into two different broad categories based mostly on whether the resulting voices are monophonic or not. The scenario wherein the resulting voices of the segregation are monophonic is titled as 'voice segregation'. On the other hand, when the resulting segments are organised in perceptually coherent groups that may include overlapping notes, then the process is referred to as 'stream segregation'. Accordingly, this work's focal point lies on stream segregation based on quantised symbolic data.

Musical content's voice/stream segregation is of great importance to Music Information Research (MIR) as it allows for efficient and higher quality analytic results, such as the identification of multiple voices and/or musical streams for the purpose of processing within the voices (rather than across voices) [2]. All in all, voice and stream segregation approaches aim at grouping notes of polyphonic musical content into entities that allow for better understanding of the underlying musical content's score [5], and for this are essential to MIR.

1.1 Motivation and Contribution

Existing methodologies of stream segregation, as extensively described in Section 2, do not utilise as many as possible of the general perceptual principles [2] that govern the way musical events are grouped together in musical streams. Moreover, previous implementations usually present low precision due to erroneous early stream assignment propagation until the end of the piece. In addition, most works of voice/stream segregation focus solely on a genre/type of musical content, thus providing genre-customised experimentation. One further setback of this genre-customised experimentation is the lack of breadth of available ground-truth for further algorithms' examination.

Accordingly, the contribution of this work is summarised as follows:

- Incorporates the general perceptual principle of *Co-*

Copyright: © 2016 Dimos Makris et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Modulation Principle that allows for ameliorated vertical integration.

- Proposes a methodology that segments musical pieces into grouping entities that allow for revision and elimination of the initial error propagation phenomenon.
- Extends the available stream segregation domain datasets with ground truth by providing new, non-pop, human-expert produced annotation of streams in musical pieces.

The rest of the paper is organised as follows: Section 2 describes background and related work and Section 3 provides a complete account of the proposed method. Subsequently, Section 4 presents and discusses the experimentation and results obtained, while the paper is concluded in Section 5.

2. RELATED WORK

Research on computational modelling of segregation of polyphonic music into separate 'voices' has lately received increased attention, though in most of these cases, 'voice' is assumed to be a monophonic sequence of successive non-overlapping musical tones.

The work of Temperley [6] proposes a set of preference rules aiming at avoiding large leaps and rests in streams, while minimising at the number of streams, avoiding the common tones shared between voices and minimising the fragmentation of the top voice. In [7], Cambouropoulos makes the case for tones being maximally proximal within streams in temporal and pitch terms, the minimisation of the number of voices and the lack of streams' crossing, i.e. the maximum number of streams to be equal to the number of notes in the largest chord. Chew and Wu [8] propose an algorithm based on the assumption that tones in the same voice should be contiguous and proximal in pitch, while voice-crossing should be avoided, i.e. the maximum number of voices to be equal to the number of notes in the largest chord. Szeto and Wong [9] present stream segregation employing a clustering modelling technique. The key assumption therein is that a stream is to be considered as a cluster since it is a group of events sharing similar pitch and time attributes (i.e. proximal in the temporal and pitch dimensions). Their algorithm determines automatically the number of streams/clusters. As aforementioned, all of these voice separation algorithms assume that a 'voice' is a monophonic succession of tones, thus focusing on the voice separation scenario.

The work by Kilian and Hoos [10] differs from the voice separation scenario as it allows for entire chords to be assigned to a single voice. Accordingly, more than one synchronous notes can potentially be assigned to one stream. Their solution segments the piece into slices with each slice containing at least two non-overlapping notes. Penalty values are used in an aggregating cost function for features that promote segregation such as large pitch intervals, rests / gaps, note overlap between successive notes, large pitch intervals and onset asynchrony within chords. The notes of each slice are separated into streams by minimisation of the cost function. The penalty values are user-adjustable in order lead to a different separation scenarios of voices by testing alternative segregation options. The maximum number of voices is again user-defined or automatically selected based on the number of notes in the largest chord. The pioneering aspect of the proposal of Kilian and Hoos lies on the fact that multi-note sonorities within single voices are allowed. Accordingly, their algorithm has a different scope/target, i.e. to split notes in different staves on a score. It takes perceptual principles in account but the result is not necessarily perceptually meaningful.

As far as the evaluation of voice/stream separation algorithms is concerned, in most of the aforementioned works, it has been performed solely on classical musical pieces. Guiomard-Kagan et. al [5] expanded their corpus to evaluate most existing voice and stream separation algorithms by adding 97 popular music pieces containing actual polyphonic information. However, the annotation used therein was based on ground truth created with monophonic voices and not streams, and thus is not applicable to our proposal.

2.1 The VISA Algorithm

The previous editions of the Voice Integration/Segregation Algorithm VISA algorithm proposed originally by Karydis et al. [11] and extended by Rafailidis et al. [3] are all based on the perceptual principles for stream separation as proposed by Bregman [12]. Basic perceptual principles, such as grouping rules based on similarity and proximity (i.e. proximal or similar entities in terms of time, space, pitch, dynamics, timbre are to be interrelated in perceptually-valid groups), have been employed in the last decades for modeling music cognition processes [13]. Huron [14] maintains that the main purpose of voice-leading in common practice harmony is to create perceptually independent musical lines/voices and presents a set of 10 perceptual principles that explain a large number of well-established voice-leading rules. The edition of the VISA algorithm proposed herein, draws on the perceptual principles presented by Huron with alterations as proposed by Cambouropoulos in [2]. The principles that are used in the previous implementations of the VISA algorithm are:

1. *Synchronous Note Principle*: Notes with synchronous onsets and same IOIs (durations) tend to be merged into a single sonority [11].
2. *Principle of Temporal Continuity*: Continuous or recurring rather than brief or intermittent sound sources' evoke strong auditory streams [14].
3. *Pitch Proximity Principle*: The coherence of an auditory stream is maintained by close pitch proximity in successive tones within the stream [14].

In order to make the distinction more clear, the original edition of the VISA algorithm as proposed by Karydis et al. in [11] is henceforth referred to as VISA07 while the edition prposed by Rafailidis et al. in [3] is denoted as VISA09.

2.1.1 Previous Editions of VISA

All editions of the VISA algorithm receive as input the musical piece in the form of a list L of notes that are sorted

according to their onset times, a window size w, and a threshold T. The output is the number V of detected musical streams. Notice that none of the VISAs demand an a-priori knowledge of the number of voices.

In detail, VISA07 and VISA09 moved in a step-wise fashion through the input sequence of musical events. The set of notes having onsets equal to the position of a "sweep line" was denoted as Sweep Line Set (SLS). Then, every SLS was divided into clusters by partitioning the notes into a set of clusters C. The clustering procedure was achieved according to the *Synchronous Note Principle*. For a set of concurrent notes at a given SLS, it had to be determined how to merge these on the set of clusters C. Since it is possible that synchronous notes may belong to different streams, VISAs examined the musical context w around these. If inside the context window, most co-sounding notes had the same onsets and offsets, implying thus a homophonic texture, then these were merged. Otherwise, this being most possibly a polyphonic texture, the notes were not merged in single sonorities. In addition, as notes with different offsets produce different clusters, each SLS was split into a number of note clusters.

In VISA07, the cluster separation was following only the *Synchronous Note Principle* while in VISA09 the *Break Cluster* module was introduced as an extra method for vertical integration. In this case, for every SLS, if the texture is homophonic and all notes have the same duration, this procedure looked ahead in the next three SLSs; if there existed more clusters in one of the following SLSs, VISA09 moved backwards and broke one by one its preceding clusters, according to the *Pitch Proximity Principle* until the current SLS cluster was examined.

Given the set of clusters C for every SLS, the horizontal streaming principle (i.e. the combination of *Temporal Continuity* and *Pitch Proximity* principles) was used to break these down into separate streams. For each SLS in the piece, a bipartite graph was formed in order to assign these to streams where one set of vertices corresponded to the currently detected streams (V) and the other set corresponded to the clusters in C. The corresponding edges represented the cost for each assignment. The cost function calculated the cost of assigning each cluster to each voice according to the *Temporal Continuity Principle* and the *Pitch Proximity Principle*.

Moreover, VISA09 included a procedure that forced the algorithm to switch onto two streams when the texture is homophonic. This was done in order not keep 'alive' extra streams (e.g. a third or fourth stream) given that the tendency was to have one or two constant streams (melody and harmonic accompaniment).

Then, using a dynamic programming technique, the best matching (lowest cost) was found between previous streams and current clusters. Finally, two additional constraints were taken into account: the former enforced stream crossing not to be allowed while the latter ensured that the top stream should be minimally fragmented [6].

2.1.2 Problems of VISA

VISA09 was tested on several musical examples that were carefully selected so as to contain a constant and small number of (up to three) streams. Most of these are homophonic pieces and the algorithm performed well in terms of precision since procedures were implemented to support better homophonic stream assignment. However, further examination showed that the algorithm's precision was diminished when tested on different music styles that contained non-homorhythmic homophonic accompanimental textures with more than 2 streams. The same phenomenon can be seen in pieces of the dataset in [3] with such homophonic texture but containing more than two streams, wherein the algorithm failed to produce a proper separation. Moreover, VISA09 was not designed to detect potential non parallel movement between notes with same onsets and offsets. Figure 1 shows an example of a non-classical piece containing non-parallel movement between notes wherein VISA09 tends to create single cluster sonorities due the homophonic texture leading to wrong stream assignment.

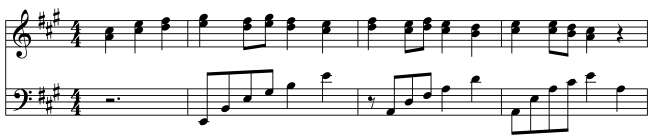


Figure 1. Excerpt from the couplet of the Greek folk song *Kaith Xwmata - Ki an se agapw den se orizw*.

In addition, the horizontal stream assignment moving by SLS from the beginning of a piece until the end can be problematic in certain cases, as the cost calculation in every SLS for assigning the streams on the current clusters is based on principles and costs of previous assignments. Therefore, if the algorithm detects in previous SLSs a wrong number of streams or clusters, it will possibly continue to accumulate wrong calculations for all the remaining SLSs even though that the piece could be very simple as far as stream assignment is concerned. This scenario was observed mainly in pieces that contain three or more streams.

Finally, the choices of the *Break Cluster* approach and the homophonic detection, which force the algorithm to switch back to the two basic streams, seem very specialised for certain (genres of) musical pieces, especially given that research for voice/stream separation has thus far mainly focused on classical music pieces.

3. THE PROPOSED METHOD

The proposed revision of VISA, the $VISA^3$ edition differs from the previous two, not only in functionality, but by additionally performing a step further after vertical integration as well as having been tested on popular music too, in addition to the common dataset of the previous two versions of VISA. We propose the use of the *Co-Modulation Principle* for further vertical integration and a customised *Contig Segmentation* approach, based on the work of Chew and Wu [8] using $clusters$. Figure 2 presents the steps of our revision which are:

1. Vertical Integration: Merging Notes into Single Sono-

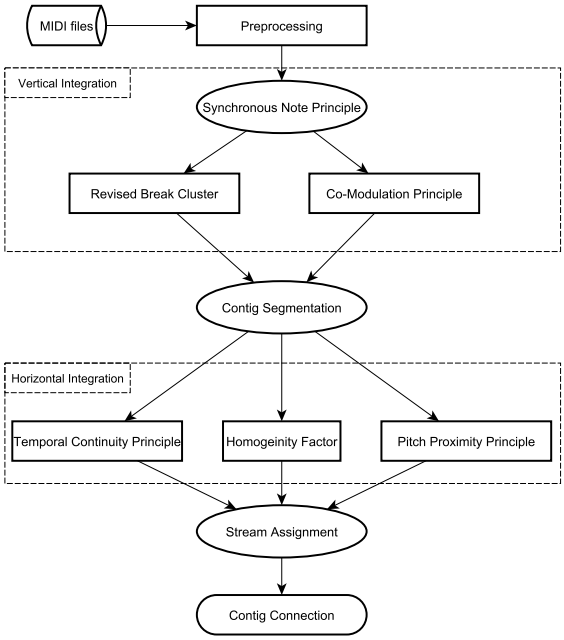


Figure 2. The $VISA^3$ algorithm.

rities using the *Synchronous Note Principle* and then examining special cases for further integration with the *Break Cluster* technique and the *Co-Modulation Principle*.

2. Contig Segmentation: Segmentation of the piece into contigs from the previous step.
3. Horizontal Integration: Stream matching within contigs using horizontal streaming principles and other factors such as homogeneity.
4. Contig Connection: Integration of contigs by connecting their streams on the segmentation boundaries.

3.1 Merging Notes into Single Sonorities

$VISA^3$ accepts as input the musical piece (i.e. a quantised MIDI file) in the form of a list L of notes that are sorted according to their onset times, a window size w and the homophony threshold T, exactly the same parameters as the previous editions of the VISA algorithm. After merging the notes into clusters according to the *Synchronous Note Principle*, further vertical integration takes place with the new revised *Break Cluster* module and the *Pitch Co-modulation Principle*.

3.1.1 Break Cluster Module

The Break Cluster module is activated when the local context is mostly homophonic and a number of notes are integrated vertically, producing thus a cluster in the current SLS. The following two significant changes occur in relation to the previous versions of the VISA algorithm:

1. Instead of looking ahead in the next three SLSs, the revised procedure of $VISA^3$ looks for the following SLSs that appear within a window size w,

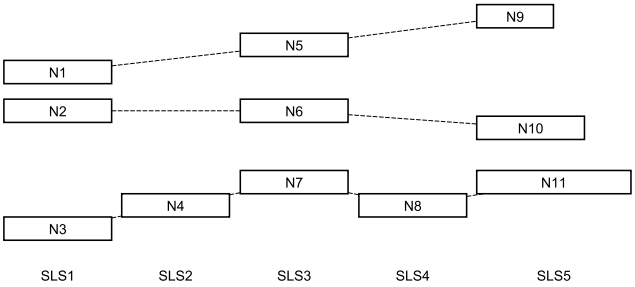


Figure 3. Breaking vertical clusters. Vertical clusters in SLS_1 and SLS_3 are broken retrospectively as the last SLS_5 comprises of three clusters; thus, this extract is separated into three streams.

2. In VISA09 the look-ahead procedure works only for single large clusters with the same number of streams, then ceases to function if it identifies more on the subsequent SLSs and starts breaking these according to pitch proximity. In $VISA^3$, the procedure doesn't stop in cases where the next SLS has less streams than the initial cluster, but it skips it and continues with the following until it finds the breaking point. In this way, the clusters that are not necessarily consecutive are being examined.

Figure 3 shows an example where the notes are in single clusters and the context is homophonic. All notes in SLS_1 are clustered vertically into a single cluster. Therefore the Break Cluster procedure is activated and looks the next SLSs in a window size w. It skips SLS_2 and SLS_4 as it detects fewer streams than SLS_1 and stops on SLS_5 as it finds three clusters: {N9}, {N10} & {N11}. Moving backwards, the process breaks SLS_3 and SLS_1 to {N5}, {N6}, {N7} & {N1}, {N2}, {N3}, respectively, based on the *Pitch Proximity Principle*. It is worth noting that if the process finds clusters with more voices than SLS_1, all combinations will be checked.

3.1.2 Pitch Co-modulation Principle

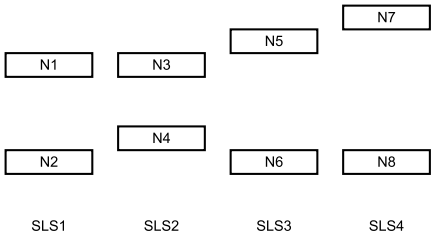
$VISA^3$ features a functionality aiming at detecting non-parallel movement between voices of consecutive vertically integrated clusters which the *Synchronous Note Principle* cannot separate. This principle is based on Huron's *Pitch Co-modulation Principle* [14]: "The perceptual union of concurrent tones is encouraged when pitch motions are positively correlated".

The procedure works as follows: In every SLS in which clusters with two or more notes are detected, it looks ahead up to a window of size w and attempts to create monophonic chains within consecutive clusters of the same number of notes. It examines whether two chains follow the same overall direction (i.e. if the notes move in parallel or not) by calculating the deviation in the pitch differences between the corresponding chain notes. Accordingly, there are two cases to be examined: two note chains in two-note cluster sequences and constant three or more note chains in three of more note clusters.

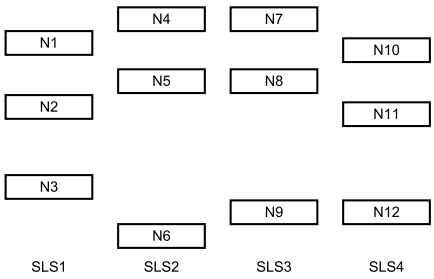
As far as the first case is concerned, the distinguishing

task is rather clear: if the concurrent notes within a chain move in non-parallel direction, these are separated and the procedure moves backwards breaking, in every SLS, the corresponding cluster into two separate clusters following the technique found in [15]. For the latter case, i.e., for larger clusters, each such cluster is separated into a set number of note chains. If the direction of notes between two chains is the same (i.e. parallel movement) then the notes of the two chains remain in the same stream. Else, if the direction of notes is different, then these form different streams. On the other hand, if there is no correlation between the movement of each stream within the chain then the cluster is separated.

The proposed methodology is based on the following two assumptions: First, the number of notes of the consecutive large clusters has to be constant. Otherwise, a cluster chain is terminated when clusters with more or less notes are found. Secondly, the direction of notes refers to the contrapuntal motion between two melodic lines [16]. While in cluster chains with two notes we seek for *parallel* motion, in this case we seek for *similar* motion, where the notion of similar motion refers to motion in the same direction. Thus, both chains move up or down but the interval between these is different in every SLS. Figure 4(a) presents examples of both cases where the notes inside the chains move in non-parallel direction and thus require separation. In Figure 4(b), the upper two streams move in parallel and thus do not require separation, in contrast to the third (lower) stream.



(a) Cluster chains with two voices



(b) Cluster chains with three or more voices

Figure 4. Examples of non-parallel movement on consecutive vertically integrated clusters.

3.2 Contig Clustering Process

The Contig Clustering process is based on the work of Chew and Wu [8] that proposed a "contig map" for voice separation. A *contig* is a collection of sequences of successive notes that belong to the same voice and the overlap depth (number of note sequences) at any time is constant. In the context of $VISA^3$, the *contig* clustering process segments a piece into contigs according to stream count and then reconnects the fragments in adjacent contigs using a distance strategy.

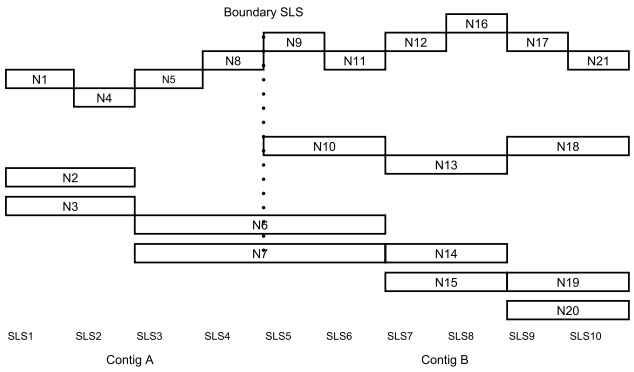


Figure 5. Contig Segmentation within a piece after vertical integration.

Thus, we propose the use of the contig mapping approach according to the cluster count as an additional step between the vertical and horizontal integration processes. Formally, if C_t represents the cluster count at SLS_t, the boundary between time slices $t-1$ and t becomes a segmentation boundary if:

1. $C_t \neq C_{t-1}$, or
2. $C_t = C_{t-1}$, in which case the cluster status changes.

The status change is caused by overlapping clusters that cross over an SLS that has been marked as a segmentation boundary. In this case, the overlapping clusters are separated at SLS_t into two clusters with the same pitch and overall duration as the initial. Figure 5 shows an example of contig segmentation. Until SLS_4 the cluster count is 2 within $Contig_A$. At SLS_5 the cluster count has not changed but an overlap cluster from previous SLS does exist. The cluster with notes {N6, N7} will be thus separated into two clusters. Thus, {N6a, N7a} will have onset as in SLS_3 and offset as in SLS_5, while {N6b, N7b} will have onset as in SLS_5 and offset as in SLS_7, respectively.

3.3 Stream Matching

As mentioned in Section 2.1.1, after determining the clusters for each SLS, a bipartite graph is created for matching notes to streams. Each cell (i, j) of the graph designates the cost between the last cluster assigned to stream i and the current cluster j. The previous versions of the VISA algorithm moved in a step-wise fashion through the input sequence, creating the graph and then assigning the streams. The following factors were used for the calculation of the cost:

1. Homogeneity factor 25%: Refers to the difference of the number of notes between clusters. Consecu-

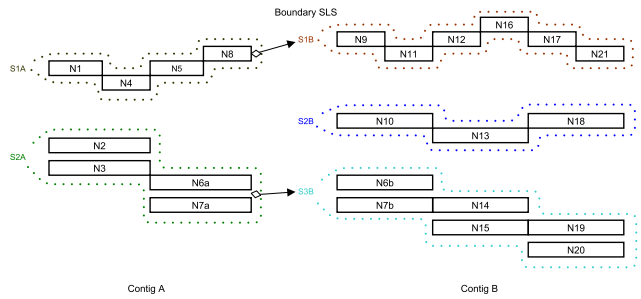


Figure 6. Stream Matching between consecutive contigs.

tive clusters with the same number of notes are more likely to belong to the same stream.

2. Pitch Proximity 50%: Distinguishes the clusters that have close average pitch with the available streams.
3. Temporal Continuity 25%: Music rests (gaps) between consecutive clusters impose additional cost for the assignment.

In $VISA^3$, we propose the same factors but with slightly different methodology:

1. Assign streams in every contig: The number of clusters C_t in a contig represents the number of streams V_t.
2. Integrate the contigs by calculating the assignment costs on all segmentation boundaries: If at SLS_t holds that $C_t \neq C_{t-1}$, then this is the end of contig Cg_{t-1} and the beginning of Cg_t. In order to connect the streams we calculate the cost using the same factors, as mentioned before, between the last clusters assigned to stream $i \in V_{t-1}$ of Cg_{t-1} and current clusters of Cg_t.

Figure 6 presents a scenario of stream assignment between contigs based on the previous example. $Contig_A$ has cluster count 2, and therefore 2 streams, $S1_a$ and $S1_b$, were assigned to all its clusters. Similarly, $Contig_B$ has 3 streams. The connection between the streams $S1_x$ and $S2_x$ is based on a stream assignment of the first clusters of $Contig_B$ with the streams of the last assigned clusters of $Contig_A$. For example, the cluster consisting of the note $N9$ is more likely to connect with a stream in which the last cluster assigned is $N8$. Therefore, a link exists between $S1_A$ and $S1_B$. Finally, it is worth mentioning that the homophonic procedure that forces the algorithm to switch to the two basic streams, as described in previous versions of the VISA algorithm, is completely removed in $VISA^3$ as it is not required due to the use of the Contig Clustering process.

4. PERFORMANCE EVALUATION

This section presents a concise description of the experimentation platform and data sets, followed by a performance analysis based on experimentation on the proposed method. The implementation is under MATLAB with the use of MIDIToolbox [17] for auxiliary functions.

4.1 Experimental Set-up

The proposed algorithm has been tested with two different datasets of quantised symbolic data. The first dataset consists mostly of the same data with the VISA09 version, for the purposes of comparing/contrasting the performance of VISA09 and $VISA^3$. It includes 30 pieces, featuring 16 excerpts primarily from piano sonatas by Beethoven, seven fugues and inventions by J.S.Bach, three mazurkas and two waltzes by F.Chopin. The selection of these pieces was intended to capture diverse musical textures, i.e. homophonic and contrapuntal textures. The majority of these pieces contain homophonic texture with two streams, consisting of a melody (upper staff) and accompanying harmony (lower staff). J.S. Bach's pieces feature independent monophonic streams, while very few pieces from Beethoven include parallel movement cases.

In order to further expand the testing corpus, we created a second small dataset with a selection of traditional Greek folk popular music. 30 MIDI files from the *Greek Music Dataset*, a freely available collection of features and metadata for 1400 popular Greek tracks [18], were selected randomly to expand the experimental examination corpus. After pre-processing, which included the deletion of duplicate instrument tracks and drum tracks, only pieces with different polyphonic and monophonic independent streams were kept. Then, an annotation task was conducted by a music theory research student that was aimed to identify streams in the scores after listening each excerpt. A number of musical examples which contained parallel movement cases, homophonic and polyphonic textures were discussed with the expert before doing this task. Therefore, bearing in mind all the above restrictions, the total number of the annotated tracks was reduced to 14.

The evaluation metric used herein is the precision of the obtained result. Herein, precision refers to the sum of notes that have been correctly assigned to the appropriate stream (according to the ground-truth), divided by the total number of notes.

4.2 Results

Table 1 shows the complete results of the proposed methodology for both datasets. The average precision of VISA09 in the classical dataset is 82,1% while with the proposed refinement, $VISA^3$ reaches 88,9%. An even more notable amelioration in precision is detected in the popular dataset where VISA09's precision is 62,8% while $VISA^3$ achieves 80,5%. Accordingly, the proposed modifications into the VISA family offer significant improvement as far as the performance of the algorithm is concerned.

More specifically, $VISA^3$ improves the precision on pieces where non-parallel movement is detected according to the *Co-Modulation Principle*, in both datasets. Accordingly, we present two examples by providing the score and the corresponding pianorolls as well as with the ground truth, for both VISA09 and $VISA^3$ assignment. Each color on the pianoroll corresponds to different stream. Figure 7 presents one such example wherein VISA09 detects two streams on the first bar, considering only the *Synchronous Note Principle*. On the other hand, $VISA^3$ detects three

	VISA09	$VISA^3$
Classical Dataset		
Beethoven, Sonata 2-1 Prestissimo	93.0%	93.6%
Beethoven, Sonata 2-1 Adagio	83.0%	86.8%
Beethoven, Sonata 2-2 AllegroVivace	79.8%	85.1%
Beethoven, Sonata 2-2 LargoApp	91.0%	95.3%
Beethoven, Sonata 2-2 Rondo	82.0%	83.9%
Beethoven, Sonata 2-2 Scherzo	75.0%	95.3%
Beethoven, Sonata 2-3 Adagio	77.0%	89.1%
Beethoven, Sonata 2-3 AllegroAssai	94.0%	98.6%
Beethoven, Sonata 2-3 AllegroConBrio	87.0%	87.3%
Beethoven, Sonata 2-3 Scherzo	73.0%	75.9%
Beethoven, Sonata 10-2 Allegretto	73.0%	90.1%
Beethoven, Sonata 10-2 Allegro	89.0%	97.2%
Beethoven, Sonata 10-2 FinalePresto	92.0%	100%
Beethoven, Sonata 13 AdagioCantabile	47.7%	78.0%
Beethoven, Sonata 13 Grave	97.9%	93.4%
Beethoven, Sonata 13 Rondo	85.0%	87.7%
Brahms, Waltz Op39 No8	89.0%	96.5%
Bach, Fugue BWV 852	91.0%	89.7%
Bach, Fugue BWV 856	94.0%	85.4%
Bach, Fugue BWV 772	96.7%	97.4%
Bach, Fugue BWV 784	93.4%	95.0%
Bach, Fugue BWV 846	49.6%	77.4%
Bach, Fugue BWV 859	32.8%	78.2%
Bach, Fugue BWV 281	39.2%	56.5%
Joplin, Harmony Club Waltz	92.3%	89.5%
Chopin, Waltz Op64 No1	91.2%	91.0%
Chopin, Waltz Op69 No2	96.2%	92.1%
Chopin, Mazurka Op7 No1	92.4%	90.8%
Chopin, Mazurka Op7 No5	96.6%	100%
Chopin, Mazurka Op67 No4	89.6%	91.3%
Popular Dataset (ID Tags)		
Marinella - Agaph pou egines dikopo maxairi ID 267	58.1%	74.4%
Marinella - Stalia, Stalia ID 10	38.1%	87.1%
Grhgorhs Bithikwtshs - Asprh Mera kai gia emas ID 385	85.6%	95.3%
Markos Vamvakarhs - Mikros Aravwniastika ID 1004	73.7%	95.1%
Mikis Theodwrakhs - Tis dikaiosynhs hlie nohte ID 1053	70.6%	81.0%
Maria Dhmhtriadh - To treno feugei stis 8 ID 1057	77.7%	88.7%
Kaith Xwmata - Ki an se agapw den se orizw ID 1240	77.6%	87.8%
Dhmhtra Galanh - Vre pws allazoun oi kairoi ID 1295	24.2%	59.9%
Vasilhs Tsitsanhs - Gia ta matia pou agapw ID 1256	65.9%	65.0%
Vasilhs Tsitsanhs - Mpakse tsifliki ID 1274	60.6%	74.7%
Vasilhs Tsitsanhs - Trekse magka na rwthseis ID 1290	32.7%	77.6%
Alikh Vougiouklakh - Gaidarakos ID 1320	71.3%	77.1%
Grhgorhs Bithikwtshs - Eimai aetos xwris ftera ID 1322	80.9%	87.2%
Mairh Lw - Epta tragoudia tha sou pw ID 1325	62.5%	75.5%

Table 1. Precision for stream separation by the previous and the current implementation of VISA on the Classical and Popular Dataset.

streams, since the top and bottom notes move in non-parallel fashion. In the second bar, both versions find 3 streams due to different note durations in every SLS while in the third bar, similarly to the case of the first bar, VISA09 detects only two of the three streams by considering solely the *Synchronous Note Principle*.

Another representative example with non-parallel movement is shown in Figure 8 where the texture can be characterised as homophonic. VISA09, when detecting homophonic texture, forces the use of one stream, i.e. all synchronized notes are assigned to one chordal stream (or two streams, i.e. main melody notes and accompaniment if melody contains some different note durations). VISA09 does not check for parallel movement in homophonic clusters and, therefore, does not have the ability to identify streams due to different motion within homophony. In this instance, it fails to recognize the three streams indicated in the ground truth, and therefore the precision is very low.

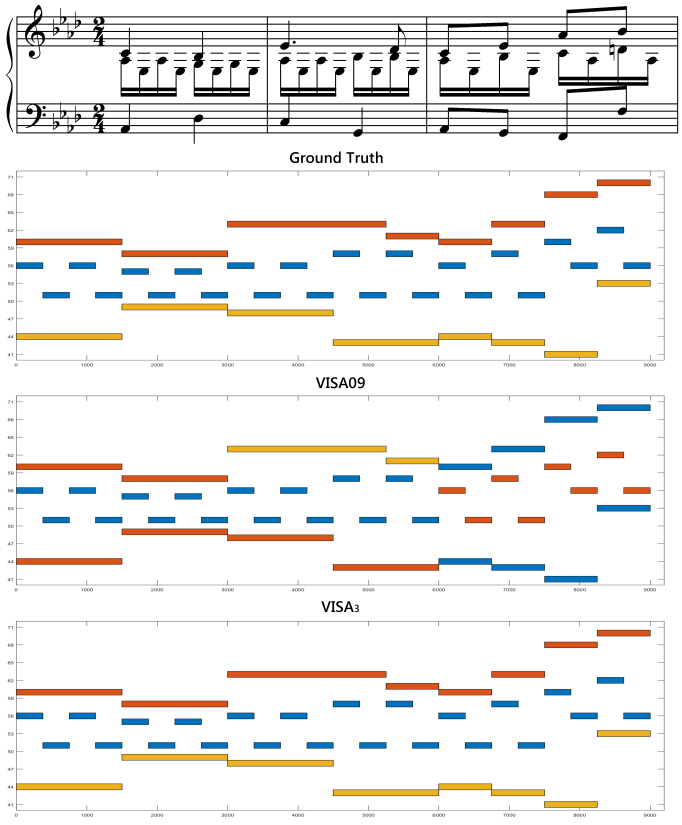


Figure 7. Opening of Beethoven's, Sonata 13, Adagio Cantabile.

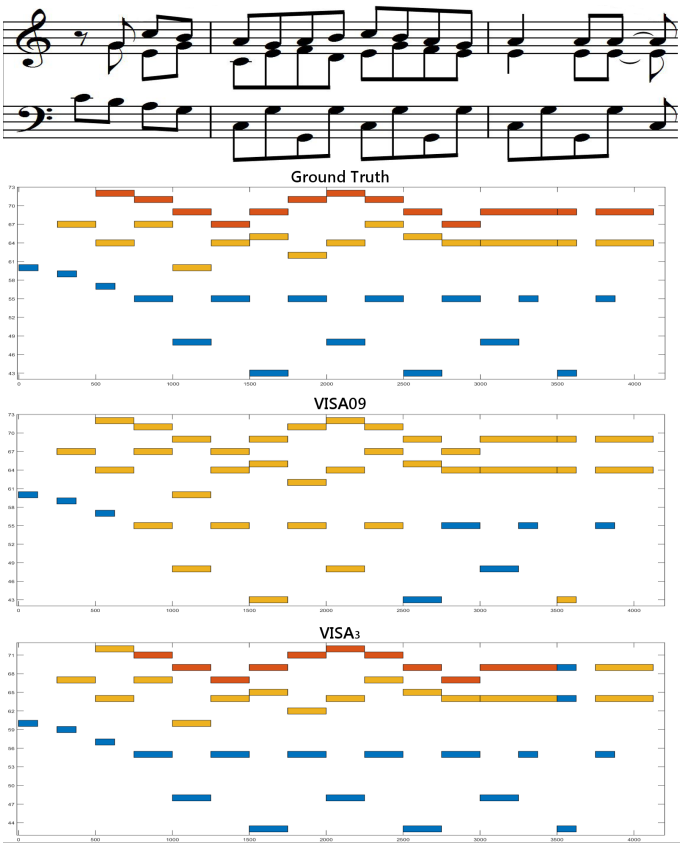


Figure 8. Opening of the Greek folk song *Alikh Vougioukklakh - Gaidarakos*.

In contrast, $VISA^3$ achieves far better results by detecting correctly the non-parallel movement between consecutive clusters and separates these to different streams. Furthermore, considering the contig segmentation of the piece, the algorithm is not carrying further initial wrong stream assignments. As shown on the assignment results for $VISA^3$ in Figure 8, $VISA^3$ fails to separate the single clusters containing two or three notes, though as the cluster count changes, a new contig begins and the stream assignment continues smoothly without taking into account previous errors.

5. CONCLUSIONS

This work presents the $VISA^3$ algorithm, a refinement of the family of VISA algorithms for integration/segregation of voice/streams. $VISA^3$ builds upon its previous editions by discarding unnecessary techniques and introducing new that adhere to general perceptual principles, address accumulation errors and tackle more generic musical content. Moreover, a new small dataset of quantised symbolic data with human-expert ground-truth annotation is utilised.

Experimental results indicated that the proposed algorithm achieves significantly better performance than its predecessors. The increase in precision is evident for both the dataset of the previous editions as well as for a new dataset that includes musical content with characteristics such that of non-parallel motion that are common and thus required to be addressed.

Future plans include the examination of alternative methods to avoid early stream assignment error propagation, less strict evaluation measurements such as customisations of the Note-based [8] and Transition-based [19] evaluation metrics used in voice separation tasks as well as and the expansion of the ground-truth dataset with more diverse musical content.

6. REFERENCES

[1] I. Karydis, A. Nanopoulos, A. Papadopoulos, E. Cambouropoulos, and Y. Manolopoulos, "Horizontal and vertical integration/segregation in auditory streaming: A voice separation algorithm for symbolic musical data," in *Sound and Music Computing Conference*, 2007, pp. 299–306.

[2] E. Cambouropoulos, "Voice and stream: Perceptual and computational modeling of voice separation," *Music Perception*, vol. 26, no. 1, pp. 75–94, 2008.

[3] D. Rafailidis, E. Cambouropoulos, and Y. Manolopoulos, "Musical voice integration/segregation: VISA revisited," in *Sound and Music Computing Conference*, 2009, pp. 42–47.

[4] E. Cambouropoulos, "'Voice' separation: theoretical, perceptual and computational perspectives," in *International Conference on Music Perception and Cognition*, 2006, pp. 987–997.

[5] N. Guiomard-Kagan, M. Giraud, R. Groult, and F. Leve, "Comparing voice and stream segmentation algorithms," in *International Society for Music Information Retrieval Conference*, 2015, pp. 493–499.

[6] D. Temperley, *The Cognition of Basic Musical Structures*. MIT Press, 2001.

[7] E. Cambouropoulos, "From midi to traditional musical notation," in *AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models of Composition, Performance and Analysis*, 2000.

[8] E. Chew and X. Wu, *Computer Music Modeling and Retrieval: Second International Symposium, CMMR 2004, Esbjerg, Denmark, May 26-29, 2004. Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, ch. Separating Voices in Polyphonic Music: A Contig Mapping Approach, pp. 1–20.

[9] W. M. Szeto and M. H. Wong, "A stream segregation algorithm for polyphonic music databases," in *Database Engineering and Applications Symposium*, 2003, pp. 130–138.

[10] J. Kilian and H. H. Hoos, "Voice separation - a local optimisation approach," in *International Conference on Music Information Retrieval*, 2002, pp. 39–46.

[11] I. Karydis, A. Nanopoulos, A. N. Papadopoulos, and E. Cambouropoulos, "VISA: The voice integration/segregation algorithm," in *International Society for Music Information Retrieval Conference*, 2007, pp. 445–448.

[12] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1990.

[13] E. Narmour, *The analysis and cognition of melodic complexity: The implication-realization model*. University of Chicago Press, 1992.

[14] D. Huron, "Tone and voice: A derivation of the rules of voice-leading from perceptual principles," *Music Perception: An Interdisciplinary Journal*, vol. 19, no. 1, pp. 1–64, 2001.

[15] D. Rafailidis, A. Nanopoulos, Y. Manolopoulos, and E. Cambouropoulos, "Detection of stream segments in symbolic musical data," in *International Society for Music Information Retrieval Conference*, 2008, pp. 83–88.

[16] R. D. Morris, "Voice-leading spaces," *Music Theory Spectrum*, vol. 20, no. 2, pp. 175–208, 1998.

[17] T. Eerola and P. Toiviainen, "Mir in matlab: The midi toolbox," in *International Society for Music Information Retrieval Conference*, 2004.

[18] D. Makris, I. Karydis, and S. Sioutas, "The greek music dataset," in *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS)*. ACM, 2015, p. 22.

[19] P. B. Kirlin and P. E. Utgoff, "Voise: Learning to segregate voices in explicit and implicit polyphony." in *ISMIR*. Citeseer, 2005, pp. 552–557.